Active Learning for Dynamical Systems

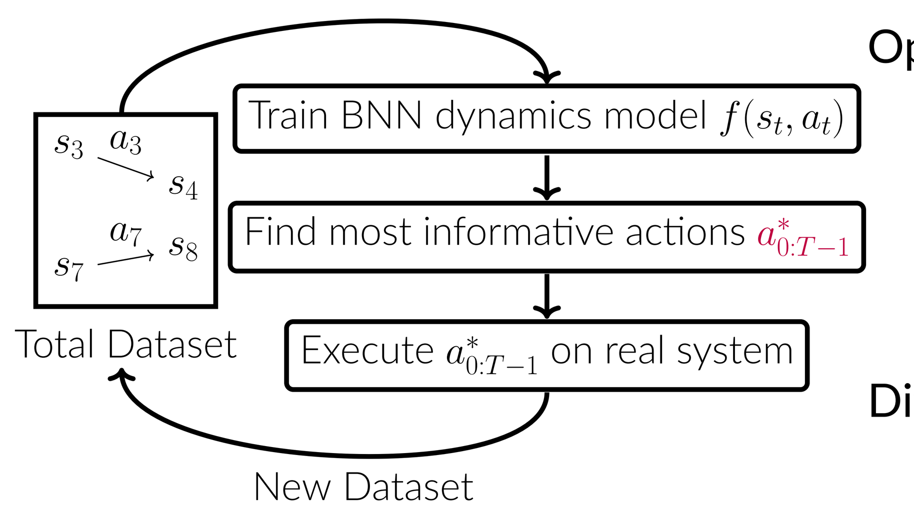
Yufei Hua yufei.hua@tum.de
Ben Sturgis ben.sturgis@tum.de

Team 10
CIT433027

Motivation and Background

- **Goal**: Learn dynamical system $s_{t+1} = f(s_t, a_t)$ using data-driven methods
- **Problem**: Collecting real-world training data is costly and time-intensive
- **Solution**: Utilize active learning methods to reduce required training data
- **Challenges in active learning for dynamical systems**:
  - System cannot be directly set to an arbitrary state
  - Identify action sequence to reach the desired state
  - Finding such sequences requires an accurate system model

Active Learning for Dynamical Systems



Optimization Problem

$$a^*_{0:T-1} = \arg\max_{a_{0:T-1}} \sum_{t=0}^{T-1} \mathbb{E}_{s_t}\left[\mathcal{H}(f(s_t, a_t))\right]$$

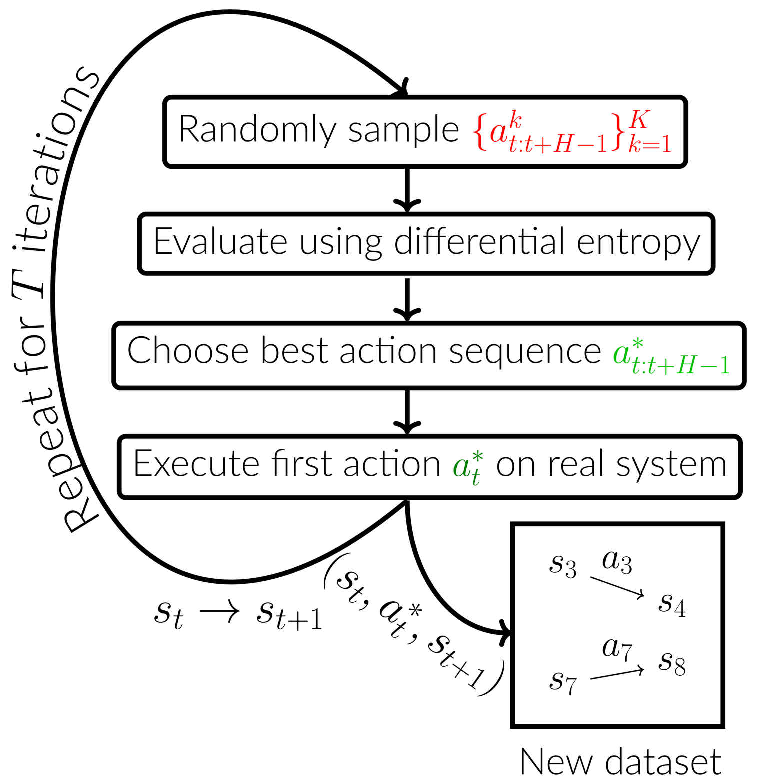
$$\text{s.t.} \quad s_{t+1} \sim p(s_{t+1} \mid s_t, a_t),$$
$$s_0 \sim p(s_0),\ a_{\min} \le a \le a_{\max}$$

Differential Entropy $\mathcal{H}(f(s_t, a_t)) \propto \ln \|\sigma_t\|_1$

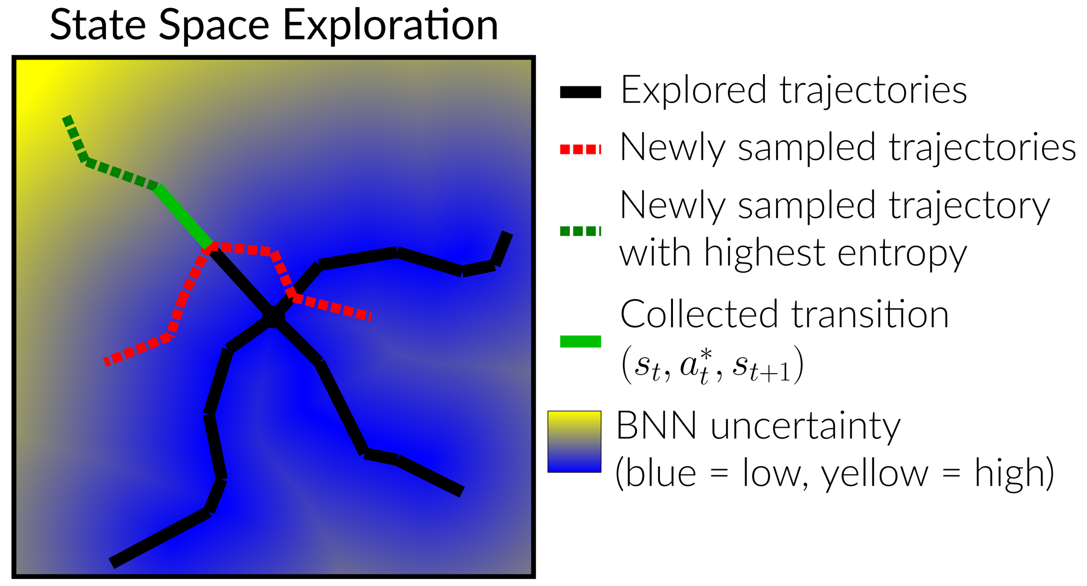
- σ_t^2: Variance of BNN model $f(s_t, a_t)$
- $\mathcal{H}$: Measure of informativeness

- T: Horizon of full trajectory

Random Sampling Shooting with Model Predicitve Control



State Space Exploration



- H: Model predictive control horizon

Soft Actor Critic

- Reward function $r_t = \mathcal{H}[f(s_t, a_t)]$ measures informativeness at time t
- Exploration-exploitation trade-off trough entropy regularization

$$\pi^* = \arg\max_{\pi} \sum_{t=0}^{T-1} \mathbb{E}_{s_t, a_t}\left[r_t + \alpha \mathcal{H}\left[\pi(\cdot|s_t)\right]\right]$$

- Off-policy algorithm $\to$ sample efficient

Challenges and Solutions

Feature Expansion for Reliable Uncertainty Estimation

- In Bayesian Neural Networks, predictive variance is proportional to the input norm $\to$ uncertainty comparisons across different norms invalid

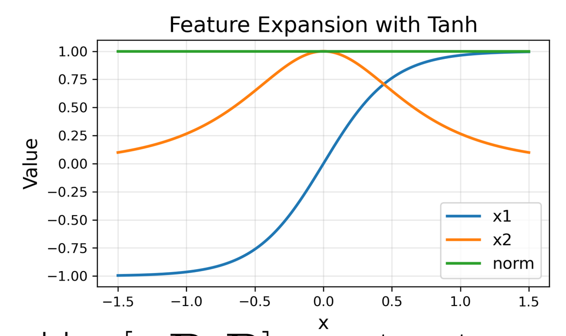
Proposition: *Consider a neural network with one hidden layer of h neurons and ReLU activation, where the parameters are initialized from $\mathcal{N}(0, \sigma^2)$. For a given deterministic input $\mathbf{x}$, Bayesian inference has the following output variance:*

$$\mathrm{Var}(y_k) = \frac{h\sigma^4}{2}\|\mathbf{x}\|_2^2$$

*Thus, the **variance** of the output **scales proportionally to** $\|\mathbf{x}\|^2$.*

Solution: Nonlinear Feature Expansion

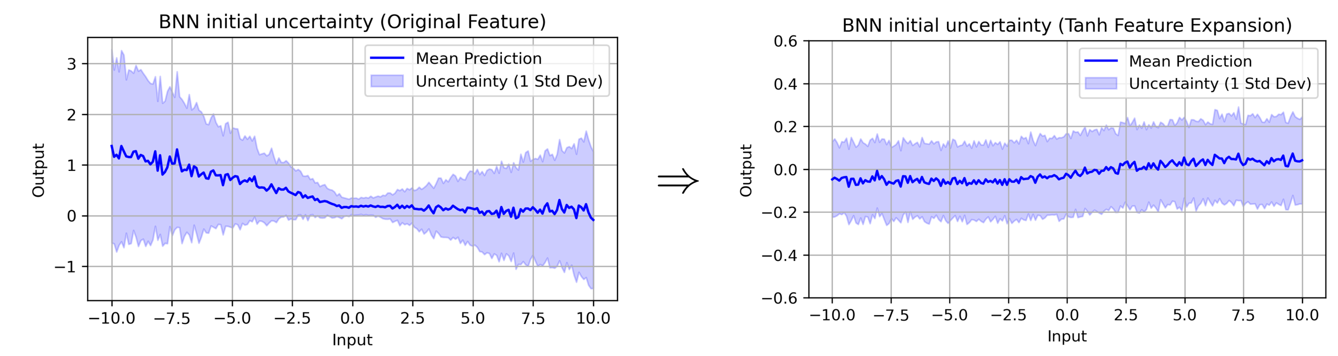
- For uniform initial uncertainty across the input space, the input norm $\|x\|$ should remain constant

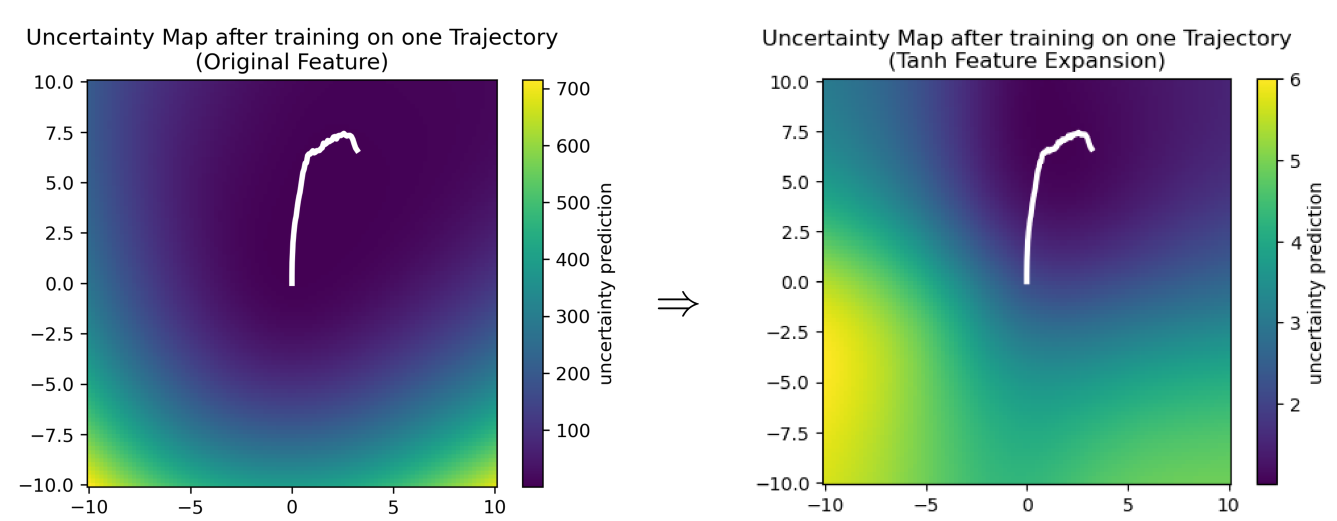


- For an original input $\mathbf{x}$ loosely constrained by $[-\mathbf{B}, \mathbf{B}]$ construct non-linear feature transformation

$$\mathbf{x}_{\text{exp}} = \begin{bmatrix} \tanh\left(\frac{2\mathbf{x}}{\mathbf{B}}\right) \\ \sqrt{1 - \tanh^2\left(\frac{2\mathbf{x}}{\mathbf{B}}\right)} \end{bmatrix}$$

ensuring that $\|\mathbf{x}_{\text{exp}}\|^2 = 1$





Uncertainty dominated by input norm — Reasonable uncertainty estimation

BNN: Laplace Approximation vs. Monte-Carlo Dropout

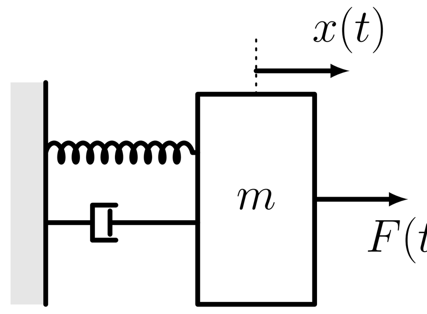
- **Monte-Carlo Dropout**: Easy to implement but sensitive to dropout probability and yields less accurate predictions
- **Laplace Approximation**: More complex to implement but allows deterministic inference like standard fully connected neural network with additional uncertainty estimation

$\to$ For small networks and dynamic learning task, **Laplace Approximation** is superior, providing accurate predictions with acceptable computational cost

Experiments and Results

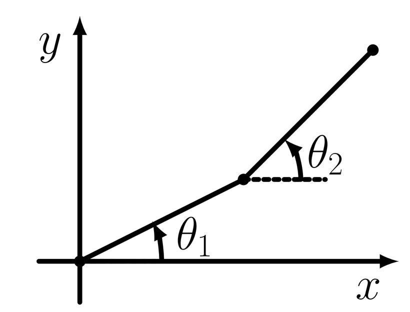
Environments

Mass-Spring-Damper



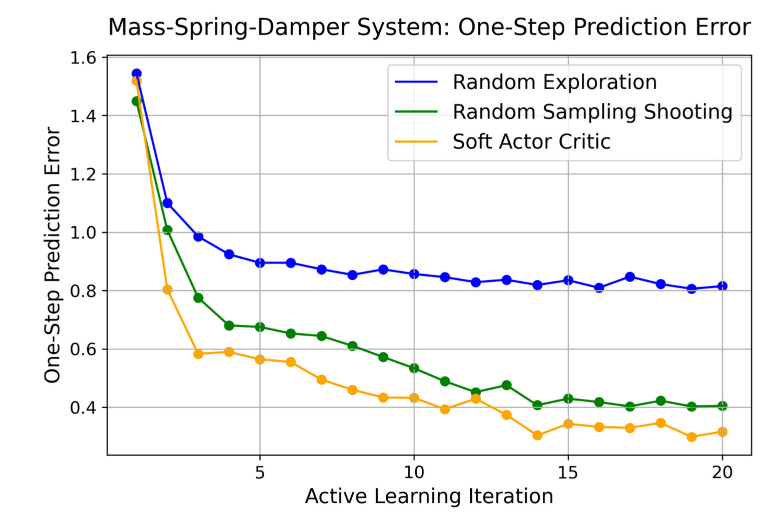
- State s: $\begin{bmatrix} x \\ \dot{x} \end{bmatrix}$
- Action a: $[F]$ with force F

Reacher

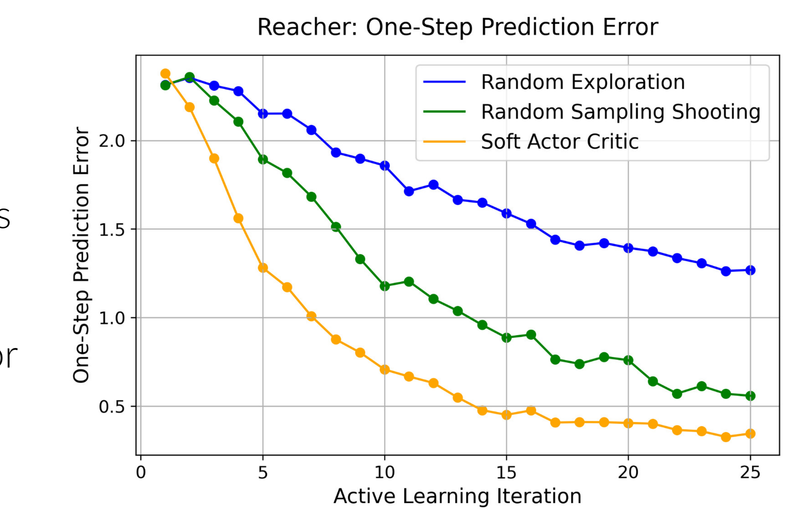


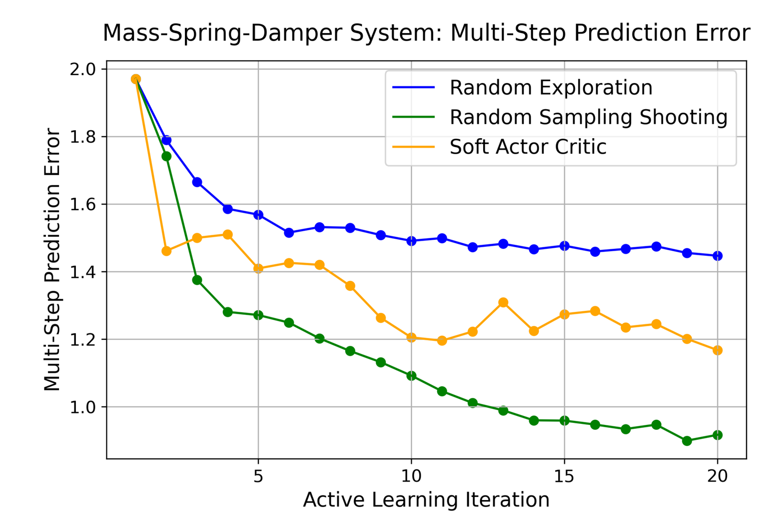
- State s: $\begin{bmatrix} \cos(\theta_1) \\ \cos(\theta_2) \\ \sin(\theta_1) \\ \sin(\theta_2) \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$
- Action a: $\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$ with torques a_1 and a_2

Prediction Error Curves

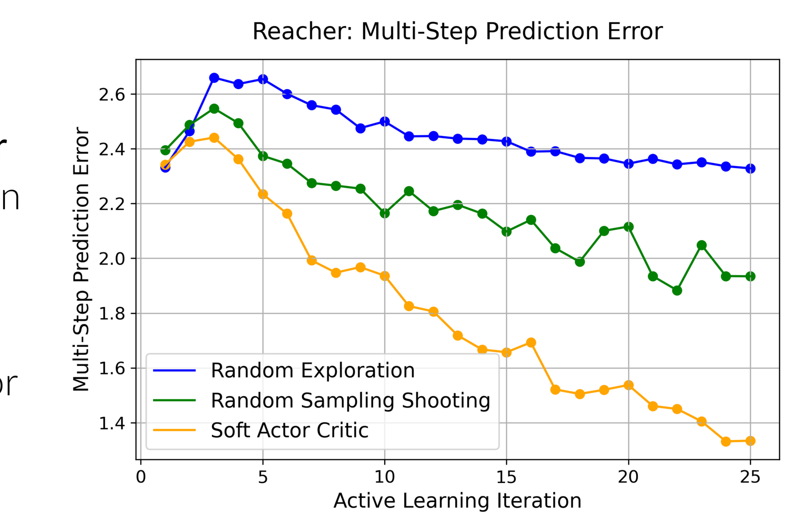


- One-Step Prediction Error
  - Sample state-action pairs
  - Perform one-step transition on real system
  - Compute prediction error





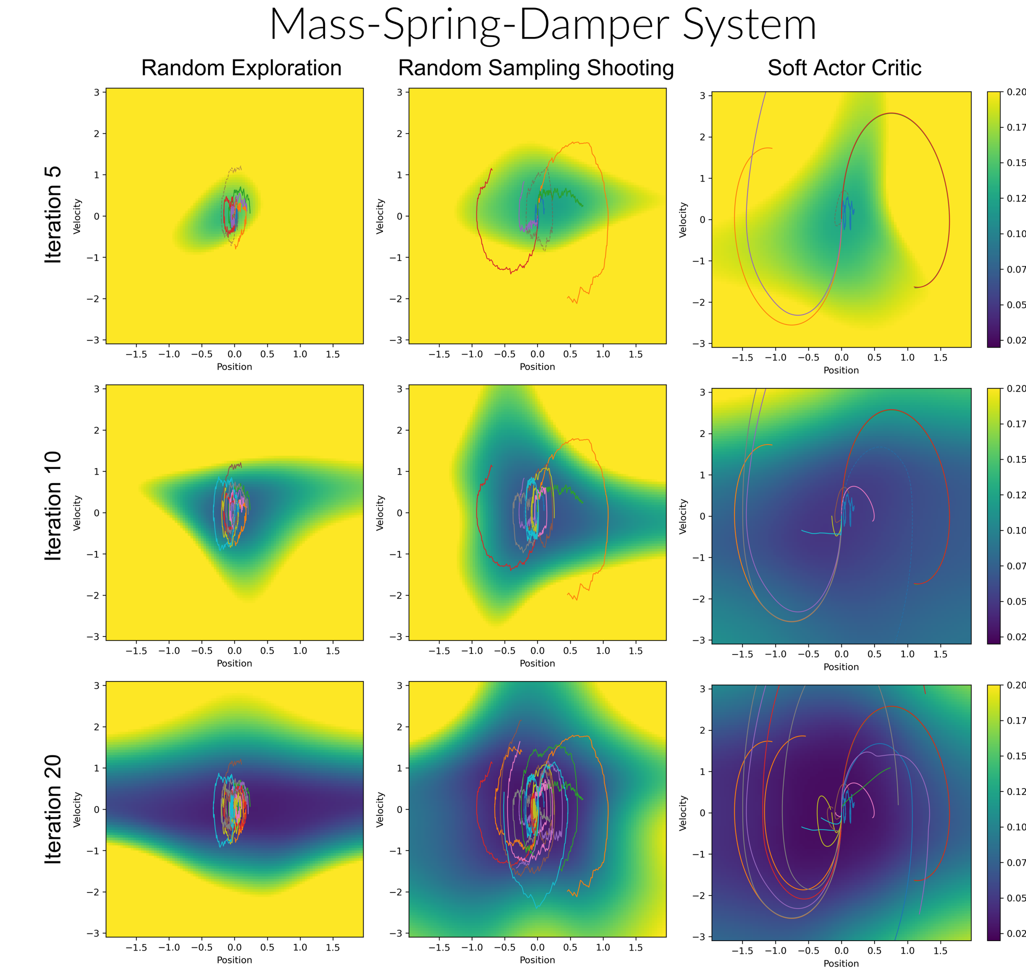
- Multi-Step Prediction Error
  - Sample initial state-action pairs
  - Perform multi-step transition on real system
  - Compute prediction error



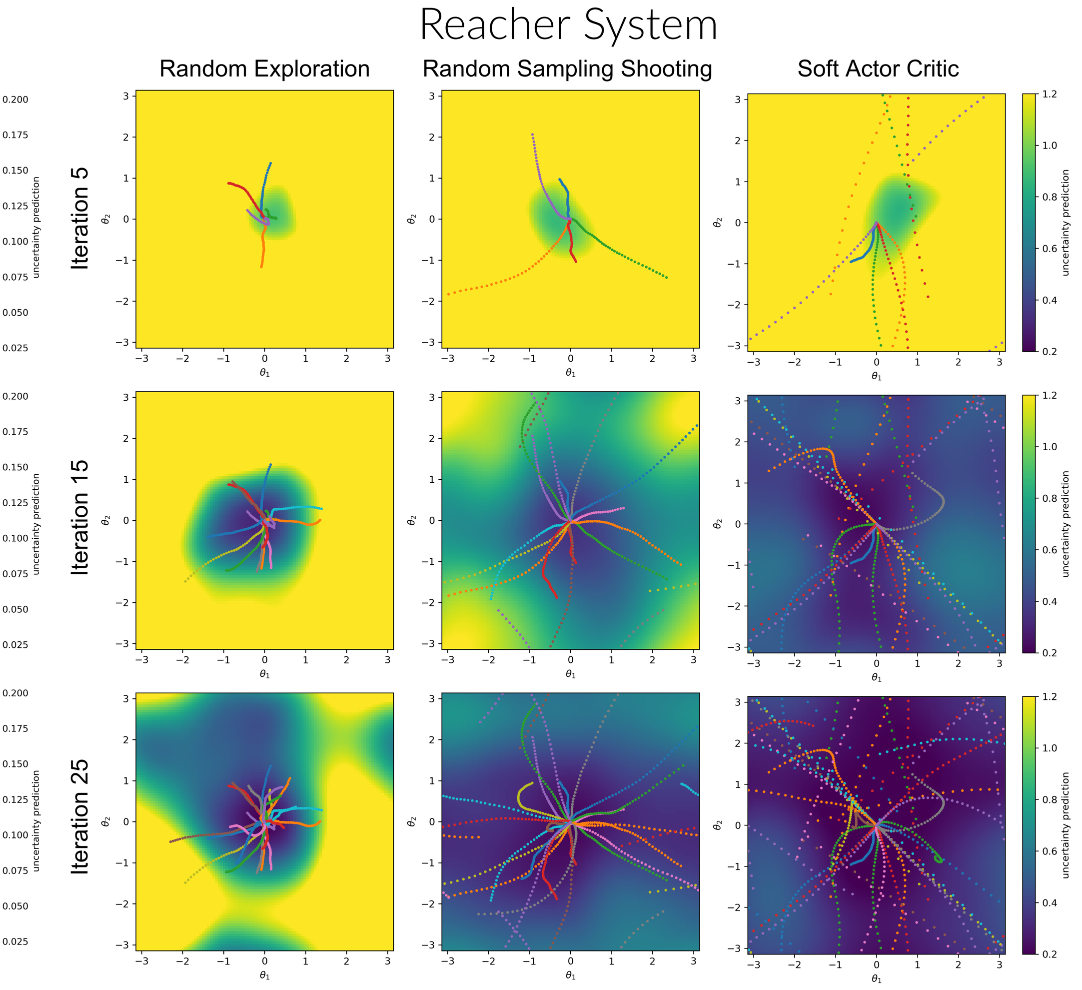
Faster error reduction: Algorithm collects more informative samples and BNN learns dynamics with less data

Uncertainty Maps

Mass-Spring-Damper System



Reacher System



- **Random Exploration**: High uncertainty remains, trajectories stay close to the origin
- **Random Sampling Shooting**: Improves state space exploration, trajectories reach out further
- **Soft Actor Critic**: Even lower uncertainty, trajectories are smoother